

**Amendments to the Specification**

Please replace the paragraph beginning on page 8, line 19, of the Specification with the following amended paragraph:

Returning to Figure 1, for any given receive/recoverable queue 107 that is populated with transaction messages from a given source node 101 and is to be used as the source of changed data to be applied to a given target node 106, Apply 108 has a browser thread 109 and one or more agent threads 112, where the number of agents is determined through user input. The work queue 110 and the done queue 111, structures internal to Apply 108, are created for the purpose of communication between the browser thread 109 and the agent threads 112. The browser thread 109 reads the transaction messages on the receive queue 107 and controls when the transaction message are placed on the work queue 110. Factors considered by the browser thread 109 include the dependencies and data constraints between the current transaction message and transaction messages already examined by the browser thread 109. Transaction messages modifying the same row can be recognized as dependents using techniques described in the co-pending U.S. Patent Application entitled, "Parallel Apply Processing In Data Replication With Preservation Of Transaction Integrity And Source Ordering Of Dependent Updates", Serial No. [ ] 10/789,775, filed on [ ] February 27, 2004. Applicant hereby incorporates this patent application by reference. The agent threads 112 apply the transaction messages on the work queue 110 in parallel with each other.

Please replace the paragraph beginning on page 14, line 17, of the Specification with the following amended paragraph:

Transaction messages modifying the same row can be recognized as dependents using techniques described in the co-pending U.S. Patent Application with Serial No. [ ]. The browser thread 109 reads incoming transaction messages from the receive queue 107 and detects any dependency with any in-flight transaction message. If no dependency is found, the transaction message is inserted into the work queue 110 from which the pool of apply agent threads 112 reads and execute the transaction messages. Once an agent thread 112 has applied a transaction message, it inserts it into the done queue 111. The browser thread 109 checks the done queue 111 for completed transaction messages and updates its dependency graph. Any transaction message without dependencies is eligible to be applied and is inserted into the work queue 110.

Please replace the paragraph beginning on page 26, line 12, of the Specification with the following amended paragraph:

A load utility is a database utility that connects to the source system, either explicitly (for example, using embedded SQL) or implicitly (for example, using nicknames) and reads all or a subset of the rows in the source table and applies them to the target table. After the RI constraints have been dropped, Apply 108 can initiate a load utility, if requested, or wait for a load utility external to Apply 108 to complete. Apply 108 then applies any row changes that may have been spilled, while the load was going on. More details on loading and spilling is further described in the co-pending U.S. Patent Application entitled, "Fault Tolerant Mechanism To Handle Initial

Load Of Replicated Object In Live System", Serial No. [-] 10/789,326, filed on [-] February 26,  
2004. Applicant hereby incorporates this patent application by reference.